# Metamorphism essay: an abstraction



Ok, let's begin with the base. If you already don't know, the metamorphism technique consists in the 'whole' change of the shape of a virus, but with- out changing its functions. Basically is what polymorphism is to the decry- ptor, but applied to a whole virus body.

Ok, ok, i know that some things weren't clear above. If you begin to think about metamorphism, you'll see that it's easy. False. It requires a great IQ, at least is what i think, because your head must calculate about all possible fails, being patient, or what will happen is that you'll never finish the goddamn virus.

One big problem is how the fuck we know what is happening with the registers in each moment. Preety easy for all the registers except for ESP... Ok, i'll discuss about ESP in another place (not here), but let me explain the other registers first. It could be done in two ways:

- know it at the same time we are morphing the code;
- or self-emulate the code.

For the first option, the implementation is preety easy, in DOS as well as in Win32. Just imagine the code of your virus that is able to generate an arithmetic operation (i.e. XOR) with an immediate. We will generate one of the morphed ways of make a XOR for the new virus body, and we will make the same XOR operation with the variable that handles the register we should modify. Same method for DOS and Win32.

The second option, in fact is more powerful, but it's very different in application matters in DOS and in Win32 enviroments. In DOS seems more easy, just because we can simply hook the interrupt 1, and trace all the virus code, knowing in all moment the content of all the registers, and at the same time, be able to generate a completly different code for obtain the same result. In Win32 the story changes, because we DON'T have our beloved interrupt 1. There isn't developed (still) any reliable tracing method for Win32 code (for viruses), anyway i think it should be done by using the Windows ability of multitask (using threads) and emulating the code at the same time it's being executed. But, as i said, this is what i think it should be done, but i haven't tested it (yet).

I know another method that will avoid us the problem of know in every moment what is happening with the registers, and i will explain it apart later.

Well, i am not sure if the title of this little part of the tute will fit what i am trying to explain perfectly, but at least it tries. I am talking about the three actual currents of undestand the metamorphism:

The first one can be defined as light metamorphism, because we only change the registers we are using along all the virus body, but anything else. So, the rate of opcodes that could be catched by the scanner is very high, and easy for them by using masks. We have an example of this technique in a virus of my friend Vecna, in his 29A age, called RegSwap.

The code permutation technique uses a more powerful technique than the above method, but for perform the same job, uses a different method, a little internal disassembler, instead the usage of tables. The example of this technique can be perfectly seen in ZCME and AZCME, by Z0MBiE (ex 29A, too). This is a light form of metamorphism too.

But the valhalla comes with the internal disassembler thingy, not used as in code permutation (for xchange registers only). We can use it for disassemble the whole virus body, and get each time what value we want to reach in each register, and perform some math operations until we have it. This is a very advanced technique, because the difficulty of its right application, and because the quality of the results. I think that one of the possible steps to follow for get the 'most hardly' detected virus is with this technique.

The perfection of metamorphism, the heaviest kind could born from the act of mixing the code permutation (exchange registers) with the internal disassembler, that is, generate each time different registers, and different ways for reach a value, by using different mathematical operations. In this way, we have the best rate of morphing under my viewpoint.

But there is still a method that i haven't listed before. I haven't done it because i will make a study about it apart, because it's the main idea for my 'Itxoiten' project.

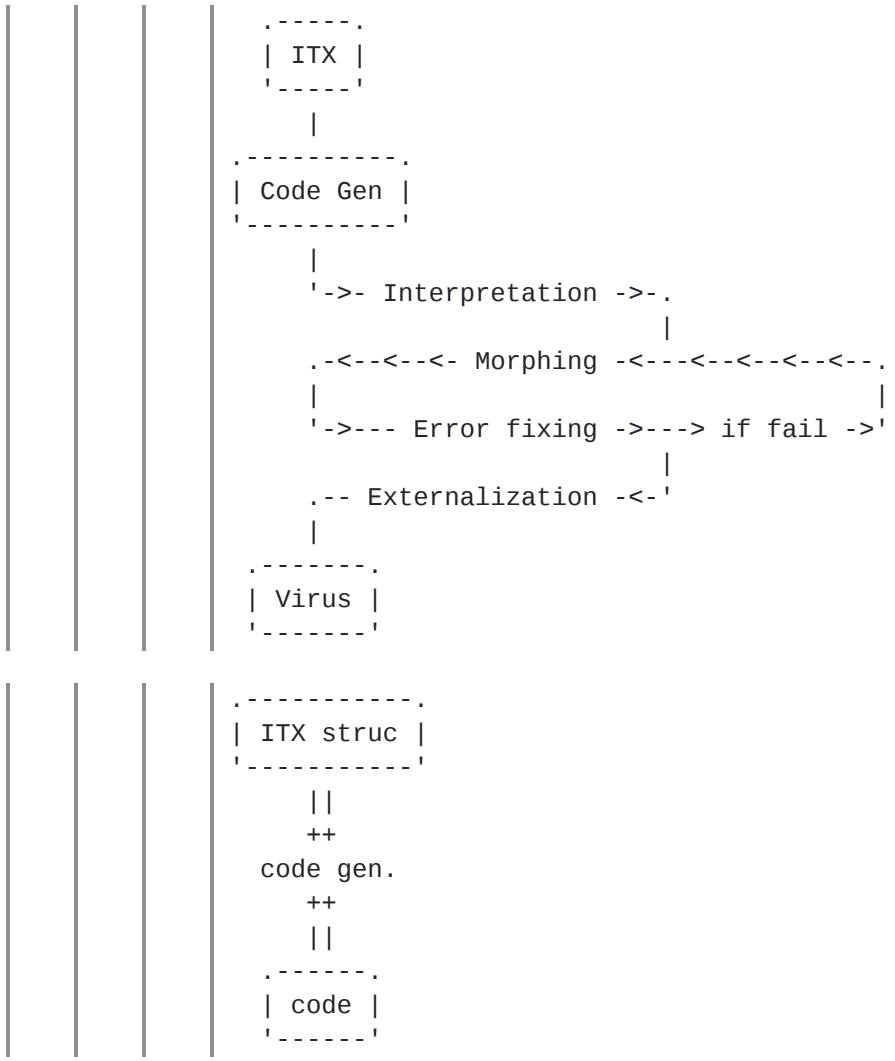
Ok, i don't know if another author have thought about this before (i think that almost all is already thought, but anyway...) but i'm gonna take my own conclussions of it.

What does 'Itxoiten' mean? Itxoiten is a word in euskera (basque country language) that means 'waiting'. And this is what you have to do until see this project finished :)

Well, it's a project of metamorphism, and probably my first virus with this technique will carry this name. Ok, before you go mad, i will explain my ideas of what am i going to do, or at least, the base concept of it.

The virus will have a code generator, but not as i said in my older article 'Viric life and die theories'. The code generator must previously make an interpretation of a structure (ITX), and generate the code indicated by it, using all the imaginable ways for do it. As you can think, this way is like to follow a map of roads, you can select many different roads to reach a point. But imagine that you can change the map. With the same engine, and another ITX structure, the code that the generator would be able to generate would be completely different. But, as you can think, this is not as easy as i exposed here. Well, that of change the structure is easy (and change it on-the-fly is too), but what isn't easy is the ITX structure. The structure must be enough powerful to handle all the situations that a Win32 virus could find in its evolution (yes, i planned 'itxoiten' for Win32 platforms).

For example, the SEH should need a non-generic part of the structure, as the code for use that technique is not very ususal (i.e. FS register usa- ge). But think, about all the actions of a Win32 (Ring-3) virus are related with the push directive, and we can use it for our own profit, playing with the register contents, and later pushing this, or that register. Or pushing the value directly, but after play with the content of ESP. Or better, both combined. Imagine the possibilities of this technique along the virus body. As the diagram situated below, our code generator should follow some basic steps.




- Interpretation, the generator analyzes properly the ITX structure, and fills up all the variables, thus initializing'em.
- Morphing, the generator makes a properly usage of the information, and it generates opcode per opcode a new virus.
- Error fixing, the generator fixes all the offsets of CALLs, unconditional jumps, conditional jumps, etc., as well as it searches for possible fails in the new generated virus code.
- Externalization, that is, write the new virus to the host, to a temp file or whatever, in its definitive version.

That are my ideas, i could change some points in the future, but basically it will remain in this way. I will work in the ITX format as well as in the generator, and i hope to finish something related to this soon.

But a new question rise now in our mind... Is worth to do a metamorphic virus (with all the effort it requires) or simply do a good polymorphic? The first opinion of anyone is that metamorphic is more powerful (it is indeed), but if you think it slowly, besides the size change (also could be reached with a good poly) we don't have many other advantages. Well, with polymorphism (of course, i'm talking about a poly a'la Mental Driller) you have fucked the AV. Metamorphism is only the polymorphism applied to a whole vi- rus, as i explained at the beginning of this document.

It should be more powerful than a simple poly, but i'd still like to make a metamorphic virus, and at least i will try to do one. It doesn't matter if the AV catches it more quickly than a virus with my latest poly engine, i'd feel better with a metamorphic beast coded by myself, and no one else :)

I really hope that you enjoyed reading this little document. I know it could be boring to you because i haven't put any practical example, as i do in my Virus Writing Guides, but sometimes my mind asks me to do something like this article. I always thought about metamorphism as the definitive weapon againist AV, without thinking about its bad points, but anyway, it's still one of our bests weapons againist the AV. Any question, suggest or correcti- ons? Mail me.